Analysis of non-product form parallel queues using a Markovian process algebra

Nigel Thomas[1] and Jeremy Bradley[2]

[1] School of Computing Science, Newcastle University, UK.
nigel.thomas@ncl.ac.uk
[2] Department of Computing, Imperial College London, UK
jb@doc.ic.ac.uk

Abstract. In this paper we use the Markovian process algebra PEPA to specify and analyse a class of queueing models which, in general, do not give rise to a product form solution but can nevertheless be decomposed into their components to obtain a scalable solution. Such a decomposition gives rise to expressions for marginal probabilities which may be used to derive potentially interesting system performance measures, such as the average number of jobs in the system. It is very important that some degree of confidence in such measures can also be given; however, we show here that it is not generally possible to calculate the variance exactly from the marginal probabilities. Hence, two approximations for the variance of the total population are presented and compared numerically.

1 Introduction

Systems of Markovian queues which give rise to product form solutions have been widely studied in the past. In this paper an alternative (non-product form) method of model decomposition is considered that can be found in the queueing network literature, *quasi-separability*. Quasi-separability was developed in the study of queueing systems which suffer breakdowns [4, 8], and generalised by Thomas et al [7, 5, 6] using the Markovian process algebra PEPA [3]. Decompositions of this kind are extremely useful when tackling models with large state spaces, especially when the state space grows exponentially with the addition of further components.

Quasi-separability can be applied to a range of models to derive numerical results very efficiently. While it does not generally give rise to expressions for joint probability distributions it does provide exact results for many performance measures, possibly negating the need for more complex numerical analysis. As such it is a very useful means of reducing the state space of large models. Not all performance measures of interest can be derived exactly from this decomposition. In particular, whilst the average number of jobs in the system may be calculated exactly, in general its variance cannot. It is clearly advantageous however, to gain some confidence in the calculated mean as a useful performance measure without having to solve a much more complicated model. Our proposed solution to this problem is to approximate the variance of the system state. Variance is

an extremely important performance measure, knowing how much a system can vary from its mean performance is an essential practical consideration. If the variation of behaviour is large then having only the mean figure for a sojourn is probably not much use for evaluation. Furthermore it has been suggested that, it in certain situations, it is more desirable for a system to be reliably predictable (more deterministic), i.e. have a low variance, rather than fast, as might be indicted by a low mean [1]. In this paper we consider a class of models consisting of a number of nodes in parallel which share a source of jobs. Each node consists of a finite length queue and one or more servers. Jobs are shared amongst the nodes on an a priori basis according to a routing vector which is dependent on the state of a scheduler. The scheduler state may change independently or in response to changes in the behaviour of the nodes. We show that if the scheduler state is not dependent on the number of jobs in the queues, then the system may be decomposed such that each node may be studied in isolation.

There are some advantages in using a process algebraic approach to tackle this problem. Firstly, the formal specification provided by the process algebra facilitates an automatic derivation of the decomposed models and therefore allows such solutions to be applied by non-experts. Secondly, we are able to explore such decompositions in a general setting in order to understand more about the properties of such models and the relationship with other possible solution methods.

In Section 2 we introduce the Markovian process algebra PEPA. In Section 3 the model is presented, followed by a PEPA representation of the model. In Section 4 discuss the decomposition and we show how mean and variance can be calculated from the marginal queue size probabilities derived. Some numerical results are presented in Section 5 for a specific example and some concluding remarks are made in Section 6.

2 PEPA

A formal presentation of PEPA is given in [3], in this section a brief informal summary is presented. PEPA, being a Markovian Process Algebra, only supports actions that occur with rates that are negative exponentially distributed. Specifications written in PEPA represent Markov processes and can be mapped to a continuous time Markov chain (CTMC). Systems are specified in PEPA in terms of *activities* and *components*. An activity (α, r) is described by the type of the activity, α, and the rate of the associated negative exponential distribution, r. This rate may be any positive real number, or given as unspecified using the symbol $\top$.

The syntax for describing components is given as:

$$P ::= (\alpha, r).P | P + Q | P/L | P \underset{\mathcal{L}}{\bowtie} Q | A$$

The component $(\alpha, r).P$ performs the activity of type a at rate r and then behaves like P. The component $P + Q$ behaves either like P or like Q, the resultant behaviour being given by the first activity to complete.

The component P/L behaves exactly like P except that the activities in the set L are concealed, their type is not visible and instead appears as the unknown type τ.

Concurrent components can be synchronised, $P \underset{\mathcal{L}}{\bowtie} Q$, such that activities in the cooperation set L involve the participation of both components. In PEPA the shared activity occurs at the slowest of the rates of the participants and if a rate is unspecified in a component, the component is passive with respect to the activities of that type. $A \stackrel{def}{=} P$ gives the constant A the behaviour of the component P. The shorthand $P||Q$ is used to denote synchronisation over no actions, i.e. $P \underset{\emptyset}{\bowtie} Q$. We employ some further shorthand that has been commonly used in the study of large parallel systems. We denote $\prod_{i=1}^{N} A_i$ to be the parallel composition of indexed components, $A_1|| \ldots ||A_N$.

In this paper we consider only models which are cyclic, that is, every derivative of components P and Q are reachable in the model description $P \underset{\mathcal{L}}{\bowtie} Q$. Necessary conditions for a cyclic model may be defined on the component and model definitions without recourse to the entire state space of the model.

3 The model

Jobs arrive into the system in a Poisson stream with rate λ. There are N nodes, each consisting of one or more servers with an associated bounded queue. All jobs arrive at a scheduler which directs jobs to a particular node according to its current state. Jobs sent to a queue which is full are lost. The system model is illustrated in Figure 1.

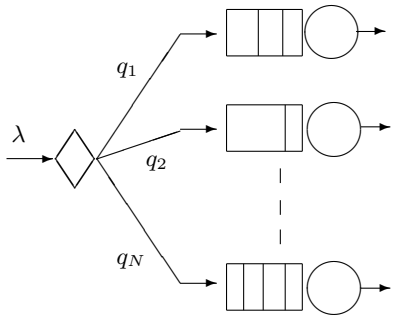

Fig. 1. A single source split among N nodes

If, at the time of arrival, a new job finds the scheduler in configuration i, then it is directed to node k with probability $q_k(i)$. These decisions are independent

of each other, of past history and of the sizes of the various queues. Thus, a routing policy is defined by specifying 2^N vectors,

$$\mathbf{q}(i) = [q_1(i), q_2(i), \ldots, q_N(i)] \ , \ i \subset \Omega_N \ , \tag{1}$$

such that for every i,

$$\sum_{k=1}^{N} q_k(i) = 1 \ .$$

The system state at time t is specified by the pair $[I(t), \mathbf{J}(t)]$, where $I(t)$ indicates the current scheduler configuration and $\mathbf{J}(t)$ is an integer vector whose k'th element, $J_k(t)$, is the number of jobs in queue k ($k = 1, 2, \ldots, N$). Under the assumptions that have been made, $X = \{[I(t), \mathbf{J}(t)] \, , \, t \geq 0\}$ is an irreducible Markov process.

We now use PEPA to specify this class of queueing system.

$$\begin{aligned}
Queue_{k,0} &\stackrel{def}{=} (arrive_k, \top).Queue_{k,1} \\
Queue_{k,j} &\stackrel{def}{=} (arrive_k, \top).Queue_{k,j+1} \\
&\quad + (service_k, \top).Queue_{k,j-1} \ , \ 0 < j < K \\
Queue_{k,K} &\stackrel{def}{=} (service_k, \top).Queue_{k,K-1}
\end{aligned}$$

$$\begin{aligned}
Scheduler_i \stackrel{def}{=} & \sum_{k=1}^{N} (service_k, \mu_{k,i}).Scheduler_i \\
& + \sum_{k=1}^{N} (arrive_k, q_k(i)\lambda).Scheduler_i \\
& + \sum_{\forall h \neq i} (switch, \alpha_{i,h}).Scheduler_h
\end{aligned}$$

$$\left(\prod_{k=1}^{N} Queue_{k,0} \right) \underset{\mathcal{L}}{\bowtie} Scheduler_1$$

Where $\mathcal{L} = \bigcup_{k=1}^{N} \{service_k, arrive_k\}$.

Clearly for this model to be irreducible we must restrict the rates of the variables $\alpha_{i,h}$ which control the switching of scheduler states, such that for each i there is at least one h such that $\alpha_{i,h} > 0$. Furthermore, for each i there must exist paths such that $\alpha_{i,a1}\alpha_{a1,a2} \ldots \alpha_{aX,1} > 0$ and $\alpha_{1,b1}\alpha_{b1,b2} \ldots \alpha_{bN,i} > 0$. That is, every scheduler state must be reachable from every other.

As we have seen, when the routing probabilities depend on the system configuration, the process is not separable (i.e., it does not have a product-form solution). As the capacity of the system becomes large, i.e. each queue has a large bound and N is also large, the direct solution becomes increasingly costly. Hence it is practically relevant to explore more efficient means of solving this class of model.

4 Quasi-Separability

A decomposition based on quasi-separability allows expressions to be derived for marginal distributions just as with a product form solution, however unlike product form these marginal distributions cannot, in general, be combined to form the joint distribution for the whole model. Despite the lack of a solution for the joint distribution, many performance measures of interest can still be derived exactly. Clearly, since exact expressions for marginal probabilities can be found, it is possible to derive any performance measure that depends on a single component. In addition it is possible to obtain certain whole system performance measures in the form of long run averages, such as the average state of the system and average response time in a queueing network.

A system that is amenable to a quasi-separable solution can be considered informally in the following way. The entire system operates within a single environment, which may be made up of several sub-environments. Several components operate within this environment such that their behaviour is affected by the state of the environment. The state of each component does not alter the state transitions of either the environment or the other components. The behaviour of such components can clearly be studied in isolation from the other components as long as the state of the environment is considered also. The restriction on the behaviour of the components imposed here is unnecessarily strong. We can also consider models where the state space of the components can be separated into that part which does have an impact on state transitions in the environment or other components and that part which has no external influence, not even on the other part of that component. Such a separation requires that the part of a component that influences the state of the environment is considered to be part of the environment for the purposes of model decomposition.

Models such as these have appeared in the literature of the study of queueing systems with breakdowns and rerouting of jobs [4, 8]. In such models the environment is generally made up of the operational state of servers in the system. For instance each server might be either working or broken, so for a system of N servers the environment has 2^N states. The routing of jobs to queues is dependent on the operational state of the system i.e. the state of the environment. Such models can generally be decomposed into single queue systems with Markov-modulated arrivals and breakdowns. This type of model is conceptually quite simple; there are only two aspects to the state of the components, but in general there may be many aspects of state that must be considered.

Consider an irreducible Markov process, $X(t)$, which consists of N separate components. The state of each component i can be described by a set of K_i separate variables. Denote by $\mathcal{V}_i$ the set of K_i variables which describe the state of component i. If it is possible to analyse the behaviour of each component, i, of the system exactly by only considering those variables that describe it, i.e. $\mathcal{V}_i$, then the system is said to be *separable*. In this case all the components are statistically independent and a product form solution exists.

For the system to be *quasi-separable* it is necessary only that it is possible to analyse the behaviour of each component, i, of the system exactly by only

considering those variables that describe it, $\mathcal{V}_i$, and a subset of the variables from all the other components. Thus the elements of $\mathcal{V}_i$ can be classified into the subsets of either system state variables, $\mathcal{S}_i$ or component state variables $\mathcal{C}_i$, such that:

- the state of $c(t) \in \mathcal{C}_i$ changes at a rate which is independent of the state of any variable $v(t) \in \mathcal{C}_j$, $\forall\, j$ such that $j \neq i$.
- the state of $s(t) \in \mathcal{S}_i$ changes at a rate which is independent of the state of any variable $v(t) \in \mathcal{C}_j$, $1 \leq j \leq N$.

If $\mathcal{C}_i \neq \emptyset$, $\forall\, i$, the system can be decomposed into N submodels such that the submodel of the system with respect to the behaviour of component i specifies the changes in the system state variables $\mathcal{S} = \bigcup_{i=1}^{N} \mathcal{S}_i$ and the component state variables $\mathcal{C}_i$. In general the analysis of these submodels gives rise to expressions for their steady-state marginal probabilities if the submodels have stationary distributions with state spaces which are infinite in at most one dimension. As stated above, these marginal probabilities do not, in general, give rise to expressions for the joint probability of the whole system, i.e. no product form solution exists. For quasi-separability to be useful the state space of the submodels should be significantly smaller than the state space of the entire model.

4.1 Deriving mean and variance from marginal probabilities

If the state space of a model is being reduced then the available information is also reduced unless a product form solution exists. The submodels consist of the system state variables $\mathcal{S} = \bigcup_{i=1}^{N} \mathcal{S}_i$ and the component state variables $\mathcal{C}_i$, hence the steady state solution of such a system gives probabilities of the form $p(\mathbf{S}, \mathbf{c}) = p(\mathcal{S} = \mathbf{S}, \mathcal{C}_i = \mathbf{c})$. A solution of the entire model would give rise to probabilities of the form $p(\mathbf{S}, \mathbf{C}) = p(\mathcal{S} = \mathbf{S}, \mathcal{C} = \mathbf{C})$, where $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$ and $\mathbf{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_N\}$. These probabilities are related in the following way for the submodel involving component i subject to the quasi-separability condition,

$$p(\mathcal{S} = \mathbf{S}, \mathcal{C}_i = \mathbf{c}) = \sum_{\forall \mathbf{C} s.t. \mathbf{C}_i = \mathbf{c}} p(\mathcal{S} = \mathbf{S}, \mathcal{C} = \mathbf{C})$$

If it is possible to associate a value, x_{ij} with each state of a component i then the average state of the component can easily be found. In addition the average of the sum of all components can be found exactly. Thus,

$$E[x_i] = \sum_{\forall j} \sum_{\forall \mathbf{S}} x_{ij} p(\mathcal{S} = \mathbf{S}, \mathcal{C}_i \equiv x_{ij})$$

Gives the average state of the component, which can be used to derive the average sum,

$$E[x] = \sum_{\forall i} E[x_i]$$

Consider, for example, the following case involving just two values:

$$E[x,y] = \sum_{i=1}^{n}\sum_{j=1}^{m}(i+j)p(i,j) = \sum_{i=1}^{n}\sum_{j=1}^{m} ip(i,j) + \sum_{i=1}^{n}\sum_{j=1}^{m} jp(i,j)$$

$$= \sum_{i=1}^{n} i\sum_{j=1}^{m} p(i,j) + \sum_{j=1}^{m} j\sum_{i=1}^{n} p(i,j)$$

$$= \sum_{i=1}^{n} ip(i,.) + \sum_{j=1}^{m} jp(.,j)$$

$$= E[x] + E[y]$$

Clearly it is an advantageous property to be able to derive system performance measures from marginal probabilities when they can be found. However, the mean is a special case as the sum of the values is trivially separated. If we consider the same example on variance the problem is evident.

$$\begin{aligned} V[x,y] &= \sum_{i=1}^{n}\sum_{j=1}^{m}(i+j)^2 p(i,j) - E^2(x,y) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}(i^2+2ij+j^2)p(i,j) - E^2(x,y) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} i^2 p(i,j) + \sum_{i=1}^{n}\sum_{j=1}^{m} j^2 p(i,j) + \sum_{i=1}^{n}\sum_{j=1}^{m} 2ijp(i,j) - E^2(x,y) \\ &= \sum_{i=1}^{n} i^2 p(i,.) + \sum_{j=1}^{m} j^2 p(.,j) + \sum_{i=1}^{n}\sum_{j=1}^{m} 2ijp(i,j) - E^2(x,y) \end{aligned}$$

In this case there is one term involving $p(i,j)$ which cannot be broken down to the marginal probabilities, $p(i,.)$ and $p(.,j)$. In the more general case where there are N components, there will be N terms involving just the marginal probabilities, but $(N-1)!$ terms involving the joint distribution. Clearly then it is not possible to calculate the variance exactly except when a product form solution exists.

The obvious (traditional) solution to this problem is to generate an approximate solution to variance by substituting $p(i,j)$ with $p(i,.)p(.,j)$, i.e. a product based approximation. In the case of quasi-separability the situation is slightly complicated since the submodels give rise to marginal probabilities involving not only component variables (as in the simple example used here), but also system state variables. The simplest solution (henceforth referred to as the *component state approximation*) would be to eliminate the system state variables by summing over all possible values:

$$p(\mathbf{c}) \approx \prod_{i=1}^{N} \sum_{\forall \mathbf{S}} p(\mathbf{S}, \mathbf{c}_i) \tag{2}$$

where $\mathbf{c} = \{\mathbf{c_1}, \ldots, \mathbf{c_N}\}$. An alternative approach (henceforth referred to as the *system state approximation*) is to attempt to derive approximations for every possible system state:

$$p(\mathbf{S}, \mathbf{c}) \approx \frac{\prod_{i=1}^{N} p(\mathbf{S}, \mathbf{c}_i)}{p(\mathbf{S})^{N-1}} \tag{3}$$

In the following section we will compare these two methods through a numerical example.

5 Example: multiple queues with unreliable servers

Now consider the following three queue example expressed in PEPA.

$$\begin{aligned}
Queue_{k,0} &\stackrel{def}{=} (arrive_k, \top).Queue_{k,1} \\
Queue_{k,j} &\stackrel{def}{=} (arrive_k, \top).Queue_{k,j+1} \\
&\quad +(service_k, \top).Queue_{k,j-1} \ , \ 0 < j < K \\
Queue_{k,K} &\stackrel{def}{=} (service_k, \top).Queue_{k,K-1}
\end{aligned}$$

$$\begin{aligned}
Scheduler_0 &\stackrel{def}{=} (repair, \eta).Scheduler_3 + (arrive_1, q_1\lambda).Scheduler_0 \\
&\quad +(arrive_2, q_2\lambda).Scheduler_0 + (arrive_3, q_3\lambda).Scheduler_0 \\
Scheduler_1 &\stackrel{def}{=} (arrive_1, \lambda).Scheduler_1 + (service_1, \mu_1).Scheduler_1 \\
&\quad +(fail_1, \xi_1).Scheduler_0 \\
Scheduler_2 &\stackrel{def}{=} (arrive_2, \lambda).Scheduler_2 + (service_2, \mu_2).Scheduler_2 \\
&\quad +(fail_2, \xi_2).Scheduler_0 \\
Scheduler_3 &\stackrel{def}{=} (arrive_3, \lambda).Scheduler_3 + (service_3, \mu_3).Scheduler_3 \\
&\quad +(fail_3, \xi_3).Scheduler_0 \\
Scheduler_4 &\stackrel{def}{=} (fail_1, \xi_1).Scheduler_2 + (fail_2, \xi_2).Scheduler_1 \\
&\quad +(arrive_1, \frac{q_1\lambda}{q_1+q_2}).Scheduler_4 + (arrive_2, \frac{q_2\lambda}{q_1+q_2}).Scheduler_4 \\
&\quad +(service_1, \mu_1).Scheduler_4 + (service_2, \mu_2).Scheduler_4 \\
Scheduler_5 &\stackrel{def}{=} (fail_1, \xi_1).Scheduler_3 + (fail_3, \xi_3).Scheduler_1 \\
&\quad +(arrive_1, \frac{q_1\lambda}{q_1+q_3}).Scheduler_5 + (arrive_2, \frac{q_3\lambda}{q_1+q_3}).Scheduler_5 \\
&\quad +(service_1, \mu_1).Scheduler_5 + (service_3, \mu_3).Scheduler_5 \\
Scheduler_6 &\stackrel{def}{=} (fail_2, \xi_2).Scheduler_3 + (fail_3, \xi_3).Scheduler_2 \\
&\quad +(arrive_2, \frac{q_2\lambda}{q_2+q_3}).Scheduler_6 + (arrive_3, \frac{q_3\lambda}{q_2+q_3}).Scheduler_6 \\
&\quad +(service_2, \mu_2).Scheduler_6 + (service_3, \mu_3).Scheduler_6 \\
Scheduler_7 &\stackrel{def}{=} (fail_1, \xi_1).Scheduler_6 + (fail_2, \xi_2).Scheduler_5
\end{aligned}$$

$$+(fail_3, \xi_3).Scheduler_4 + (arrive_1, q_1\lambda).Scheduler_7$$
$$+(arrive_2, q_2\lambda).Scheduler_7 + (arrive_3, q_3\lambda).Scheduler_7$$
$$+(service_1, mu_1).Scheduler_7 + (service_2, \mu_2).Scheduler_6$$
$$+(service_3, \mu_3).Scheduler_6$$

$$(Queue_{1,0}||Queue_{2,0}||Queue_{3,0}) \underset{\substack{service_2, arrive_3, service_3\}}}{\overset{\{arrive_1, service_1, arrive_2}{\bowtie}} Scheduler_7$$

This model represents three queues whose servers suffer independent failures and subsequent repairs. A repair will repair the entire system, but will only be triggered once all the servers have failed. The scheduler attempts to route jobs to active servers, if any exist. In the case of all the servers being broken ($Scheduler_0$) the scheduler routes jobs to all queues in the same proportion as if all were working.

Clearly this model fits the decomposition class introduced in the previous section. The number of jobs in each queue is not dependent on the number in the other queues, however, all queue lengths are dependent on the behaviour of the scheduler component. Thus we can decompose this model into three smaller ones, defined by the following system equations.

$$Queue_1 \underset{service_1\}}{\overset{\{arrive_1,}{\bowtie}} Scheduler_7$$
$$Queue_2 \underset{service_2\}}{\overset{\{arrive_2,}{\bowtie}} Scheduler_7$$
$$Queue_3 \underset{service_3\}}{\overset{\{arrive_3,}{\bowtie}} Scheduler_7$$

Each of these models has $8(K+1)$ states in the underlying CTMC, whereas the original model has $8(K+1)^3$ states. Obviously if K is large, then this is a considerable saving, possibly meaning that the decomposed models are numerically tractable when the full model is not.

The immediate advantage of this decomposition is that we can quickly find global average metrics as described above, which can then be used to optimise parameters. In the case of this example we can numerically optimise the routing probabilities q_k to minimise the average number of jobs in the system or the average response time. Performing optimisations of this kind on the whole model would be extremely costly.

5.1 Numerical results

We now turn our attention to the problem of estimating the variance of the total number of jobs in the system. For this exercise we will assume that the three servers are identical, meaning of course that the optimal (static) routing probabilities will be equal, i.e. $q_k = \frac{1}{3}$. In all cases the queues were bounded at $K = 10$. We will then investigate how the two approximations perform as we vary the load and the duration of repair periods.

Figure 2 shows the relationship between variance and load. At low load the variance is low, as the number of jobs in any queue rarely grows very large. As the load increases, so does the variance, until leveling off and then decreasing due to the effect of the bound. The variance decreases at high load as the queues become full most of the time. As can be seen, with these parameters, both approximations work well.

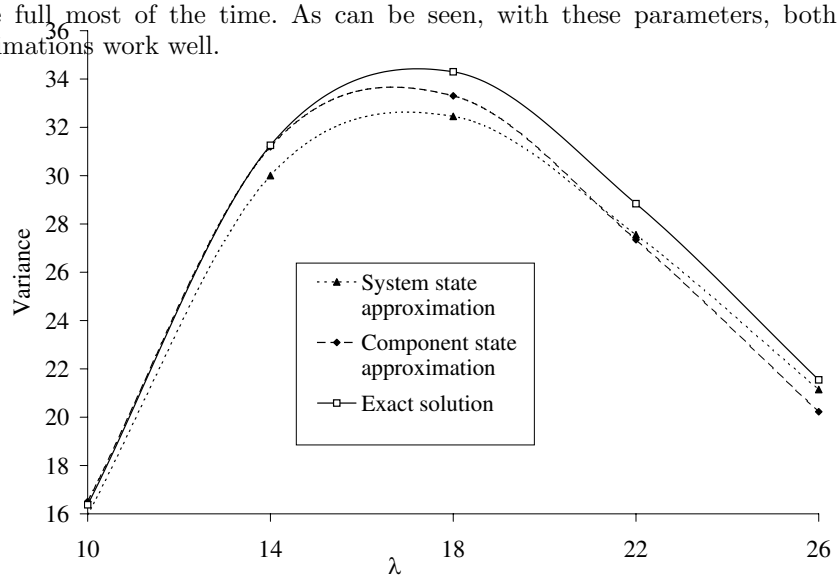

Fig. 2. Variance of the total number of jobs against arrival rate
$\mu_k = 10$, $\eta = 10$, $\xi = 1$, $q_k = \frac{1}{3}$

Figure 3 shows the variance as a function of the failure rate, ξ_k. The repair rate is also varied in direct proportion to the failure rate, so that the probability of being in any given scheduler state is the same for each value of ξ_k. When the failure (and repair) rate is relatively large the interruptions to service are relatively brief and so few arrivals occur when all the servers are broken. However, when the repair rate is decreased, the duration period for which all servers are broken increases and so the queue will fill up. Thus, when the repair rate is small, there becomes a big difference between the queue lengths in $Scheduler_7$ and the queue lengths in $Scheduler_0$. As the repair rate continues to decrease, this difference does not increase any more as the queues cannot exceed their bound, hence the variance levels off.

In Figure 3 there is much less correspondence between the approximations and the exact result. Apart from $\xi_k = 0.1$, there is fairly good correlation between the system state approximation and the exact result. In other examples we have observed that when the approximations closely agree, they are accurate, however, that does not hold here when $\xi_k = 0.1$.

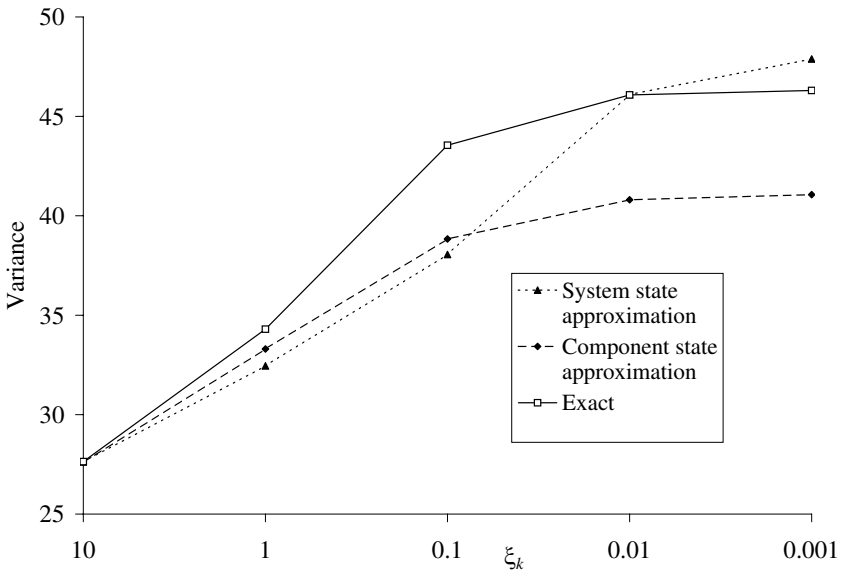

Fig. 3. Variance of the total number of jobs varied with failure rate $\lambda = 18$, $\eta = 10\xi$, $\mu_k = 10$, $q_k = \frac{1}{3}$

6 Conclusions

In this paper we have shown how a class of queueing model can be specified using PEPA and formally decomposed into a number of submodels. These submodels are easier to solve numerically, but have the weakness that it is not possible to derive the joint queue length probabilities exactly. As a consequence we have investigated two approximations for the joint queue length probability which can be used to predict the variance of the total population.

The approach has been illustrated through a significant example. This has shown that there is a huge potential saving in computational effort through this method. However, computing the approximations is not trivial and their accuracy is not universal. Therefore, the main lesson is that this decomposition is mainly useful as a way of obtaining metrics which are based entirely on the marginal queue length probabilities. Estimates of other metrics are clearly useful and particularly so if there is a very high cost of obtaining an exact solution.

Acknowledgements

Both authors are supported by the EPSRC funded AMPS project, which is held jointly at Imperial College London (EP/G011737/1) and Newcastle University (EP/G011389/1).

References

1. J. Bradley and N. Davis, Measuring improved reliability in stochastic systems, in: *Proceedings of 15th UK Performance Engineering Workshop*, pp. 121-130, University of Bristol, 1999.
2. G. Clark, S. Gilmore, J. Hillston and N. Thomas, Experiences with the PEPA performance modelling tools, *IEE Proceedings - Software*, **146**(1), 1999.
3. J. Hillston, *A Compositional Approach to Performance Modelling*, Cambridge University Press, 1996.
4. I. Mitrani and P.E. Wright, Routing in the Presence of Breakdowns, *Performance Evaluation*, **20**, pp. 151-164, 1994.
5. N. Thomas, Extending Quasi-separability, in: *Proceedings of 15th UK Performance Engineering Workshop*, pp. 131-140, University of Bristol, 1999.
6. N. Thomas and J. Bradley, Approximating variance in non-product form decomposed models, in: *Proceedings of the 8th International Workshop on Process Algebra and Performance Modelling*, Carleton Scientific Publishers, 2000.
7. N. Thomas and S. Gilmore, Applying Quasi-Separability to Markovian Process Algebra, in: *Proceedings of 6th International Workshop on Process Algebra and Performance Modelling*, 1998.
8. N. Thomas and I. Mitrani, Routing Among Different Nodes Where Servers Break Down Without Losing Jobs, in: *Quantitative Methods in Parallel Systems*, pp. 248-261, Springer-Verlag, 1995.